

8

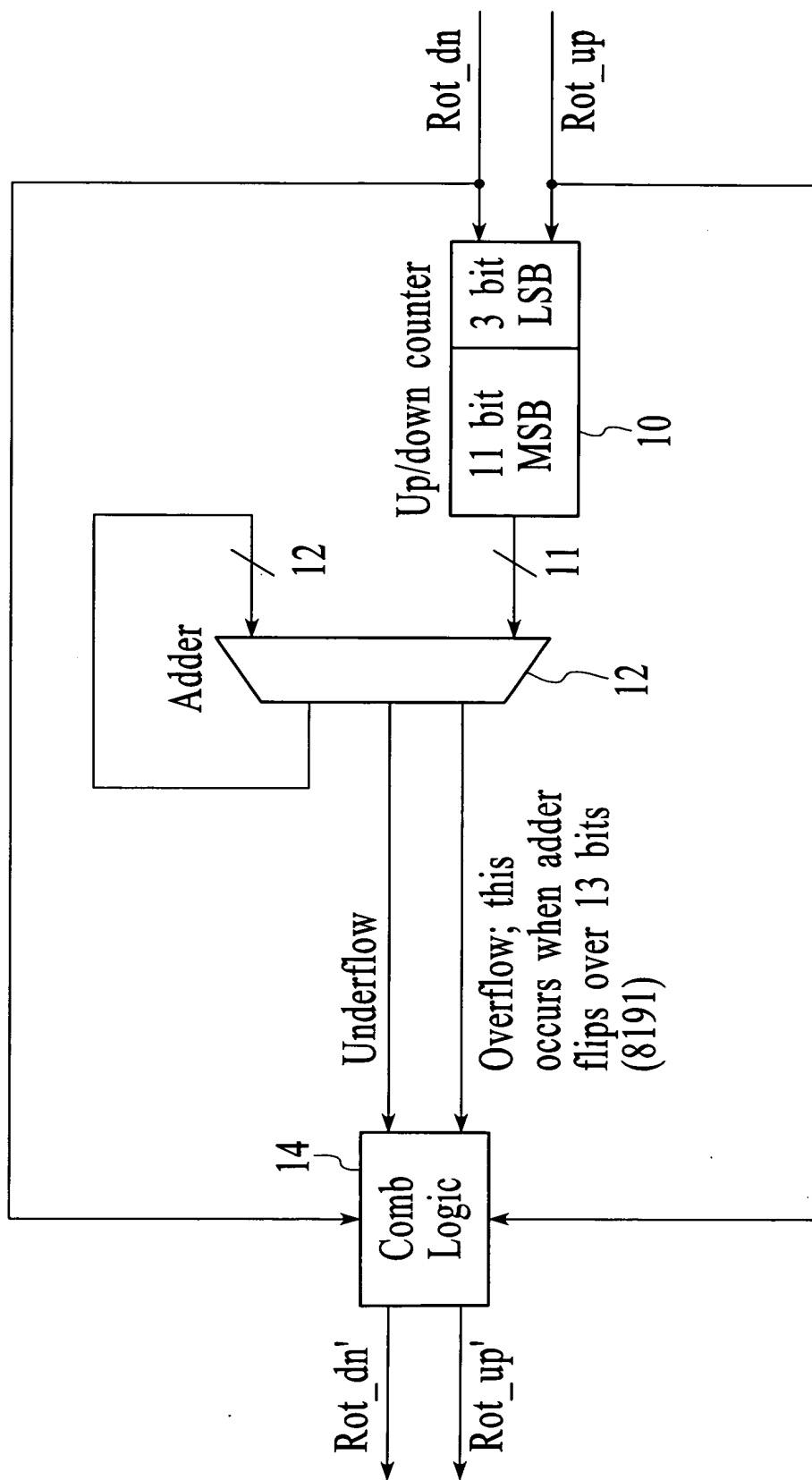


FIG. 1

C++ Code from system simulation:

```
//now check if the fly wheel is active
if (FlyWheelActive)
{
    FlyWheelUp[FlyWheelRateCounter] = "OutUp;
    FlyWheelDown[FlyWheelRateCounter] = "OutDown;
    if (++ FlyWheelRateCounter == 4) // FlyWheel runs at 1/16 rate
    {
        FlyWheelRateCounter = 0; // Reset 0,1,2,3 counter
        // extract the up/down overhang
        Sum = 0;
        for (i=0;i<4;i++)
            Sum += FlyWheelUp[i] - FlyWheelDown[i];
        if (Sum > 0) FlyWheelCounter++;
        if (Sum < 0) FlyWheelCounter--;
        Div_result = div (FlyWheelCounter,AverageFactor);// average over the 3 LSBs
        FlyWheelAdder += Div_result.quot;
        Div_result = div (FlyWheelAdder*192);// test for overflow over 2^13
        if (Div_result.quot>= 1)
        {
            *OutUp = false;
            *OutDown = false;
            FlyWheelAdder = Div_result.rem;
            Sum = *OutUp + true = *OutDown - false;
            If (Sum > 0) *OutUp = true;
            If (Sum < 0) *OutDown = true;
        };
        If (Div_result.quot<=-1)
        {
            *OutUp = false;
            *OutDown = false;
            FlyWheelAdder = Div_result.rem;
            Sum = *OutUp + false = *OutDown - true;
            If (Sum > 0) *OutUp = true;
            If (Sum < 0) *OutDown = true;
        };
    };
};
```

<- check if fly wheel is switched on

<- transformation of early/late to quarter rate

<- check for early or late overhang

<- step the counter accordingly up or down

<- Average over the LSB

<- Add the counter MSBs to the accumulator

<- Check for positive accumulator overflow

<- Generate an 'up' signal for positive overflow

<- Check for negative accumulator overflow

<- Generate a 'down' signal for positive overflow

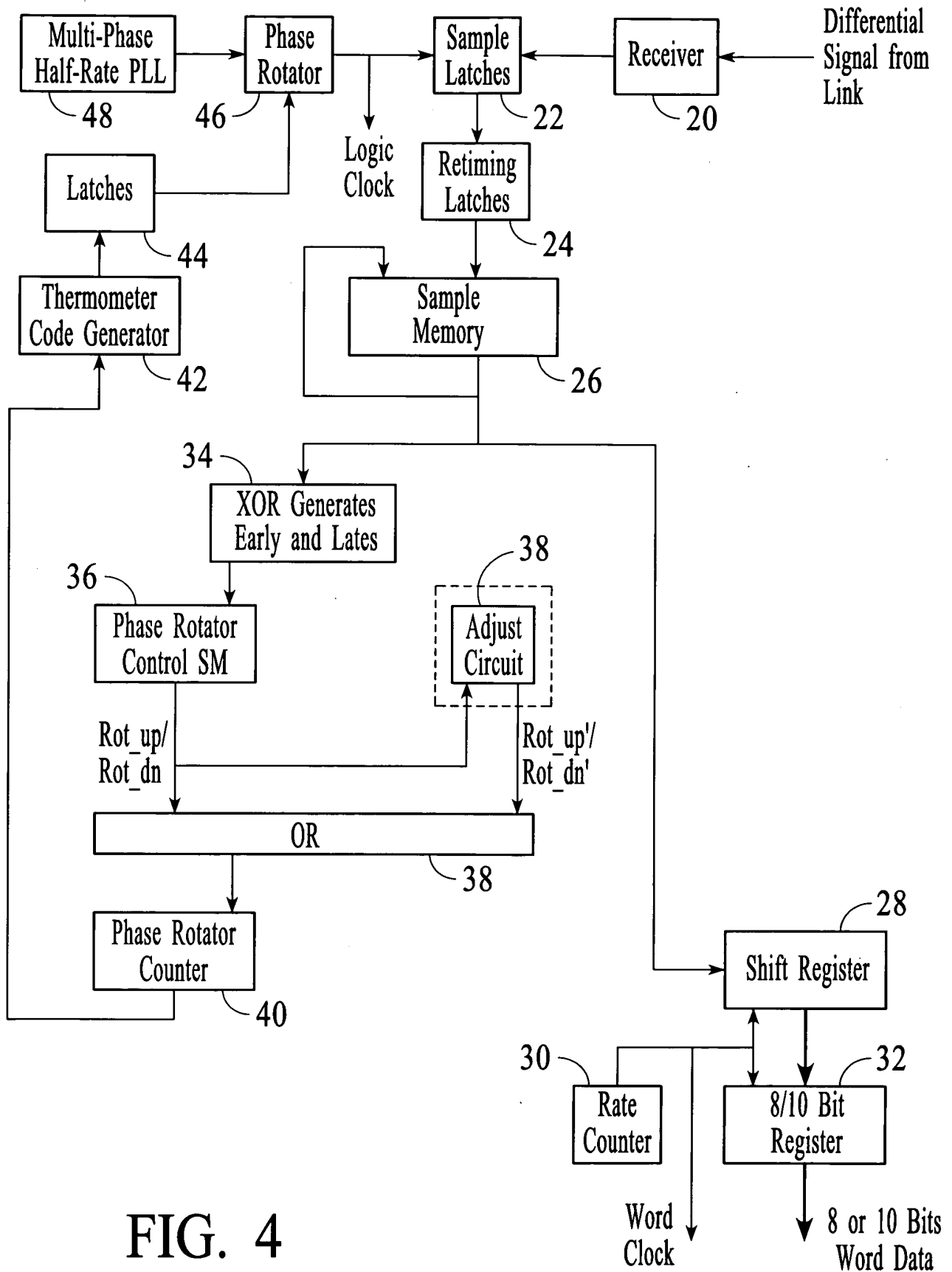


FIG. 4